Score-based Continuous-time Diffusion Models

Lecture 9

18-789

Logistics

- Wednesday: Project proposal presentation
- Also Wednesday: HW2 due!
- Next week: spring break (no class)

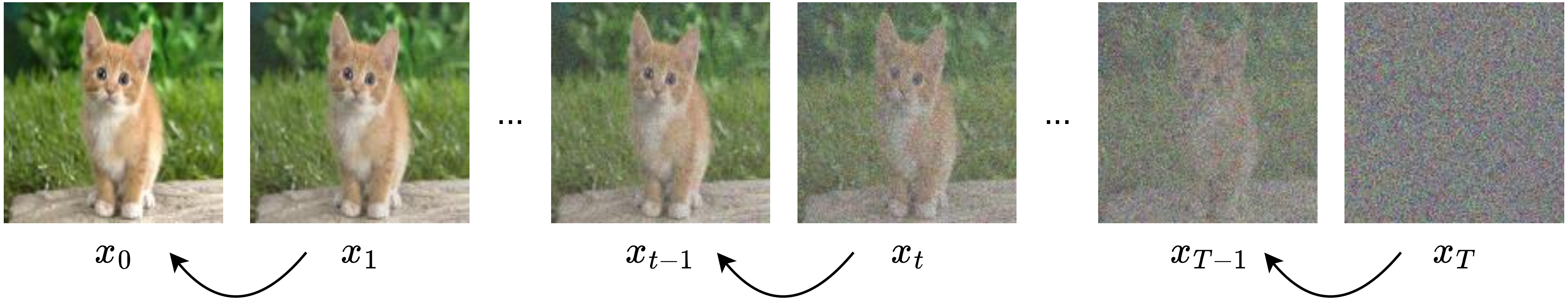
Logistics

- Next next Wednesday
  - Student Presentation III: **Hybrid** Deep Generative Models
  - Special instruction: try to cover the following key questions
    - Significance: **How** do they **combine** different generative models?
    - Significance: **Why** do they need to combine them? Why can't they just use one of them?
    - Limitation: Are we **losing something** when combining multiple types of models?
    - Limitation: Are there **better ways to combine**?
  - Task, data, evaluation, etc. are secondary (mention them very briefly unless they are related to the key questions!)
  - Related work (cover only 1-2)

Recap: Diffusion Model is a Denoising VAE

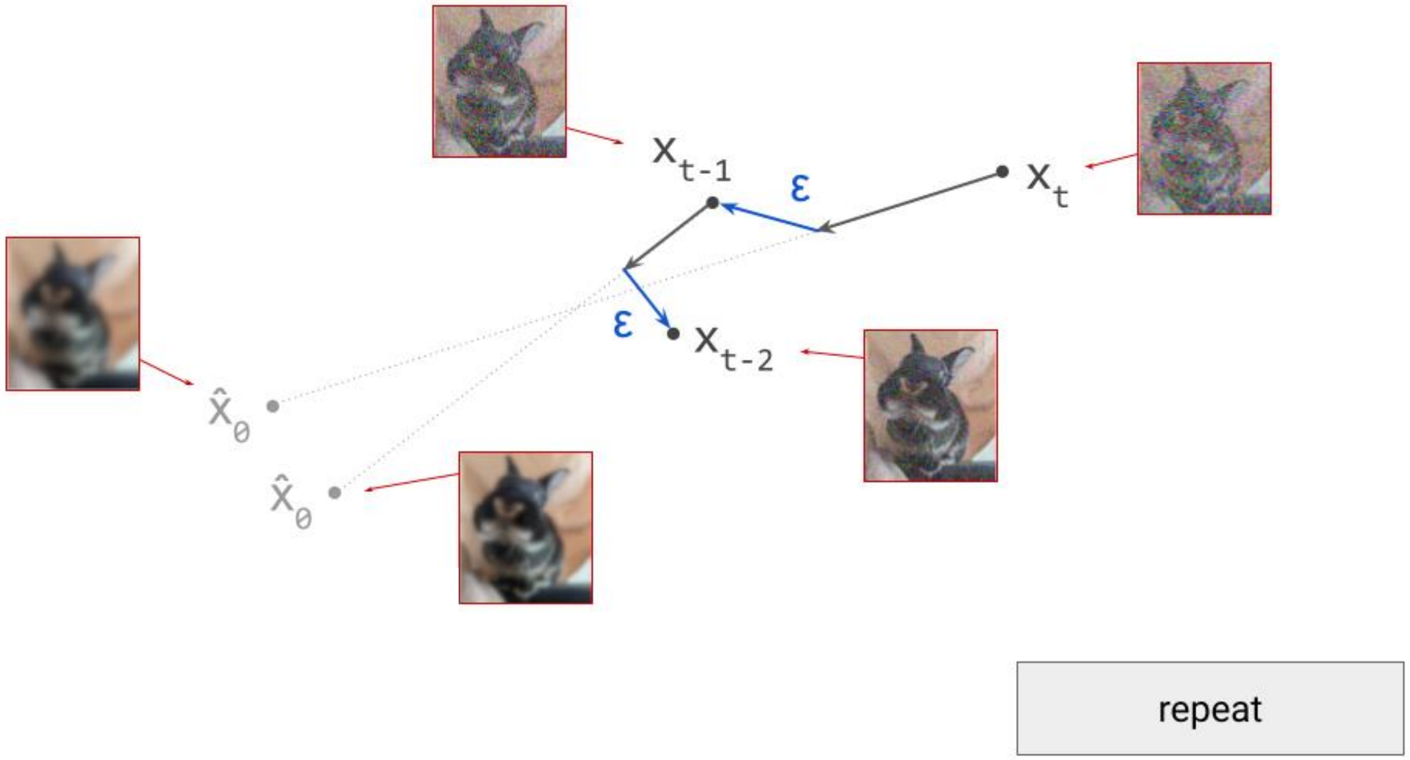
- Training: Denoising objective
- Inference: Starting from pure noise, iteratively remove noise

$$x_t = \sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon$$



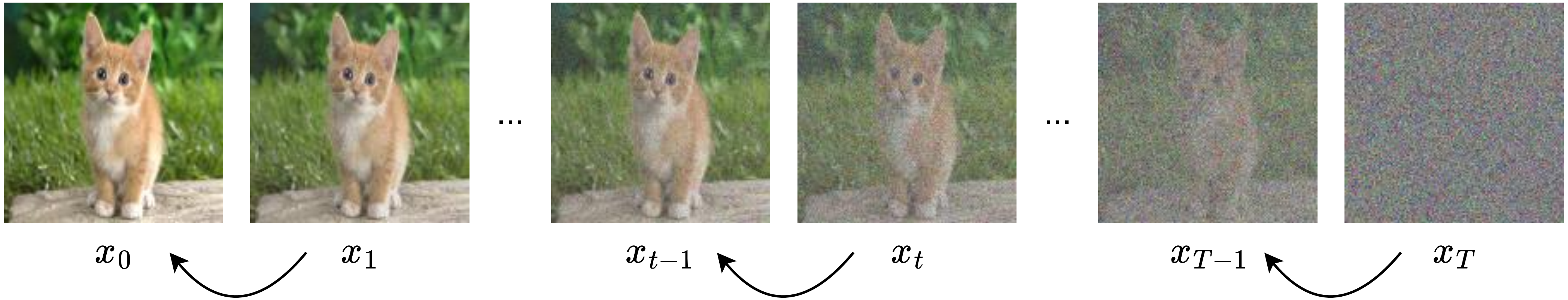
Recap: Diffusion Model is a Denoising VAE

- Training: Denoising objective
- Inference: Starting from pure noise, iteratively remove noise



Recap: Diffusion Model is a Denoising VAE

- Training: Denoising objective
- Inference: Starting from pure noise, iteratively remove noise
- Three equivalent prediction targets
  - $\tilde{x}_{t-1}, x_0, \epsilon$
  - Mathematically **equivalent**, but empirically **not the same** (as training targets)

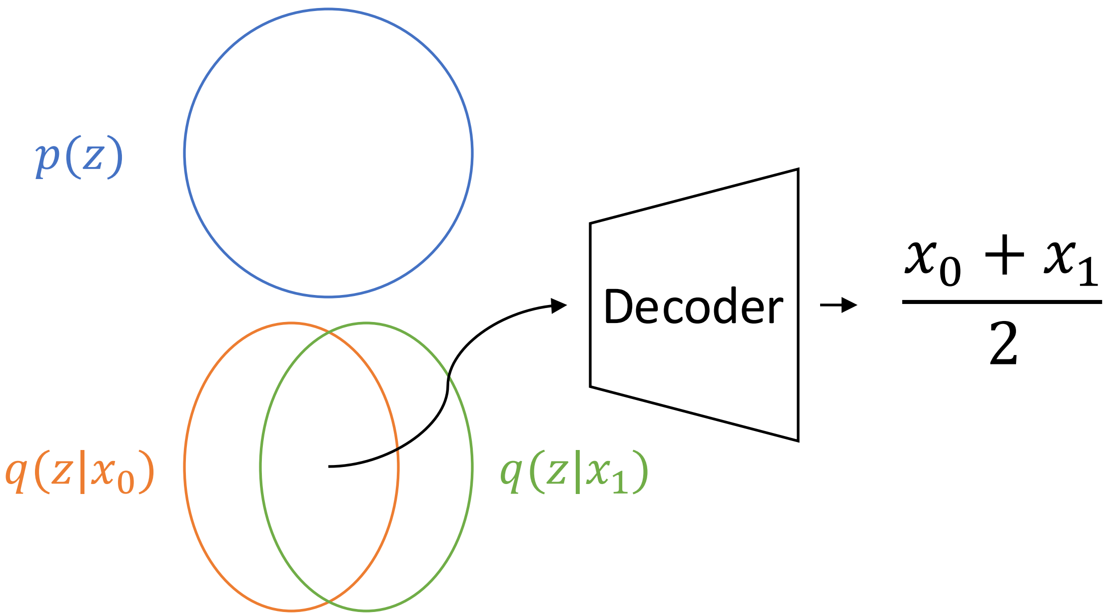


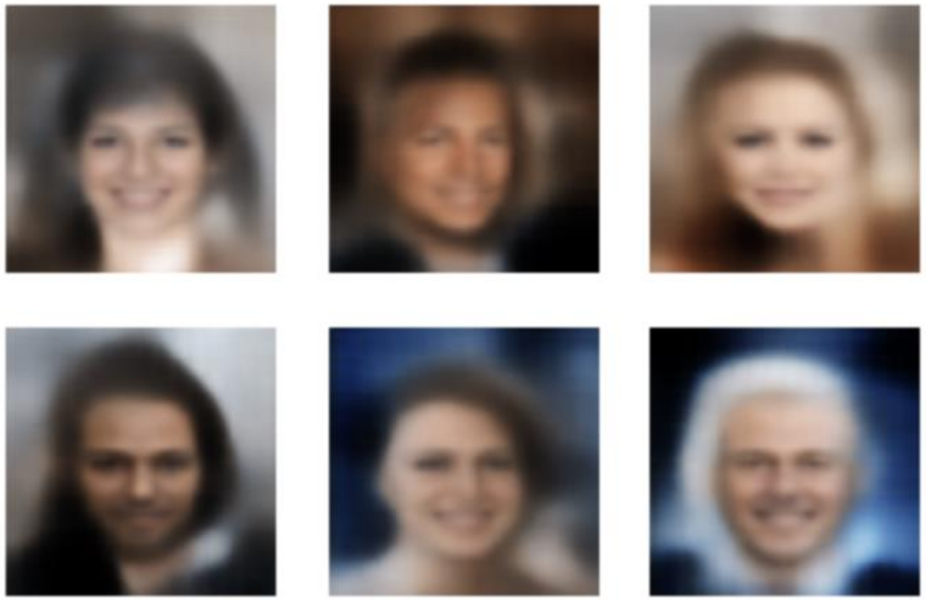
Recap: Diffusion Model is a Denoising VAE

- Training: Denoising objective
- Inference: Starting from pure noise, iteratively remove noise
- Three equivalent prediction targets
  - $\tilde{x}_{t-1}, x_0, \epsilon$
- Connection to VAE

Why doesn't Diffusion generate blurry images (like VAE)?

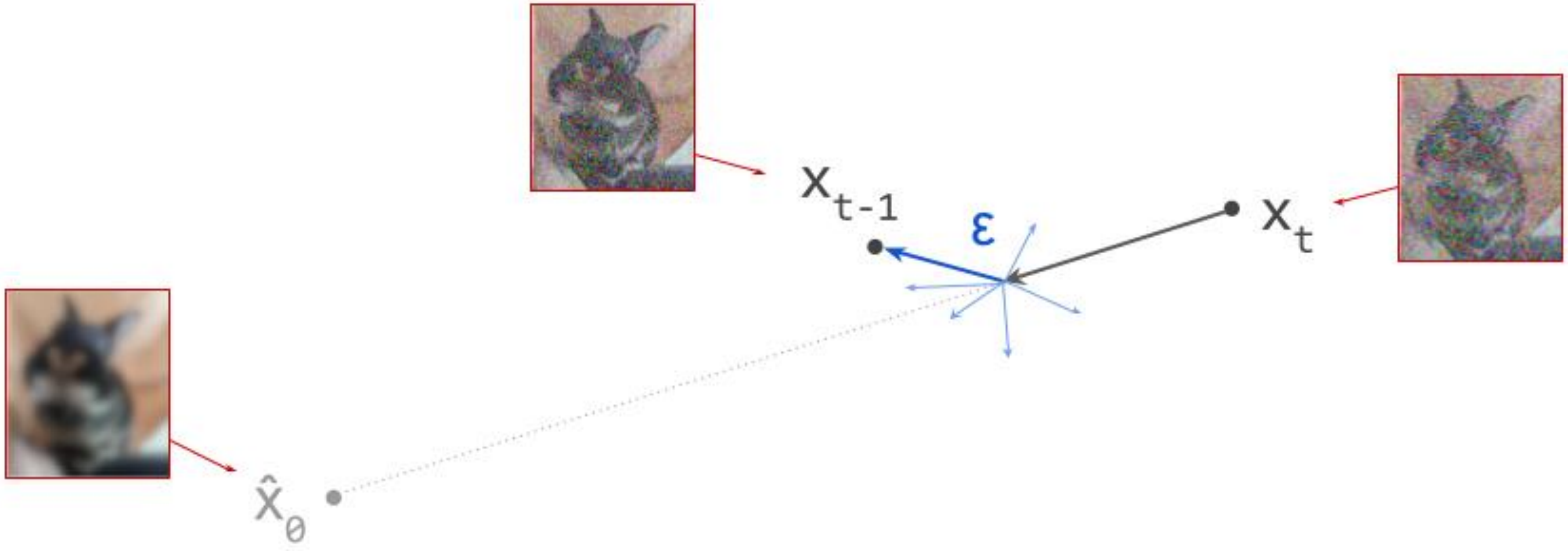
- VAE samples are blurry because the decoder must "average" over all plausible outputs compatible with the latent code.

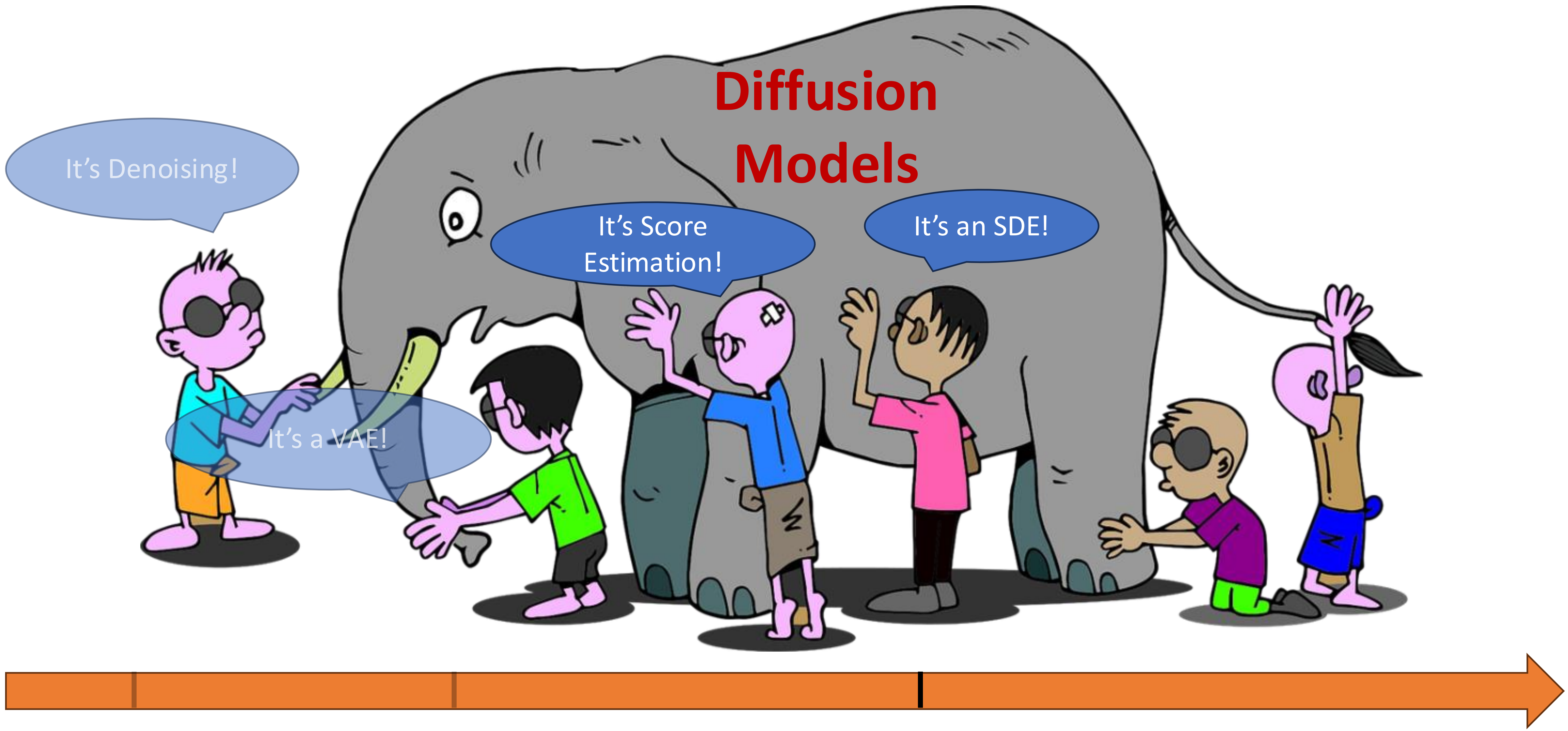




Why doesn’t Diffusion generate blurry images (like VAE)?

- In Diffusion Models, the x_0 prediction is also blurry (for the same reason)!
- BUT: we do not use intermediate x_0 predictions as the final outputs.
  - When each step is small enough, $q(x_{t-1}|x_t)$ is almost deterministic
  - Only a single possible x_{t-1} that can produce x_t





"Deep Unsupervised Learning using Nonequilibrium Thermodynamics" 2015

"Denoising Diffusion Probabilistic Models" 2020

"Score-Based Generative Modeling through Stochastic Differential Equations" 2021

What is “score”?

- Score: gradient of log-likelihood $s(x) = \nabla_x \log p(x)$

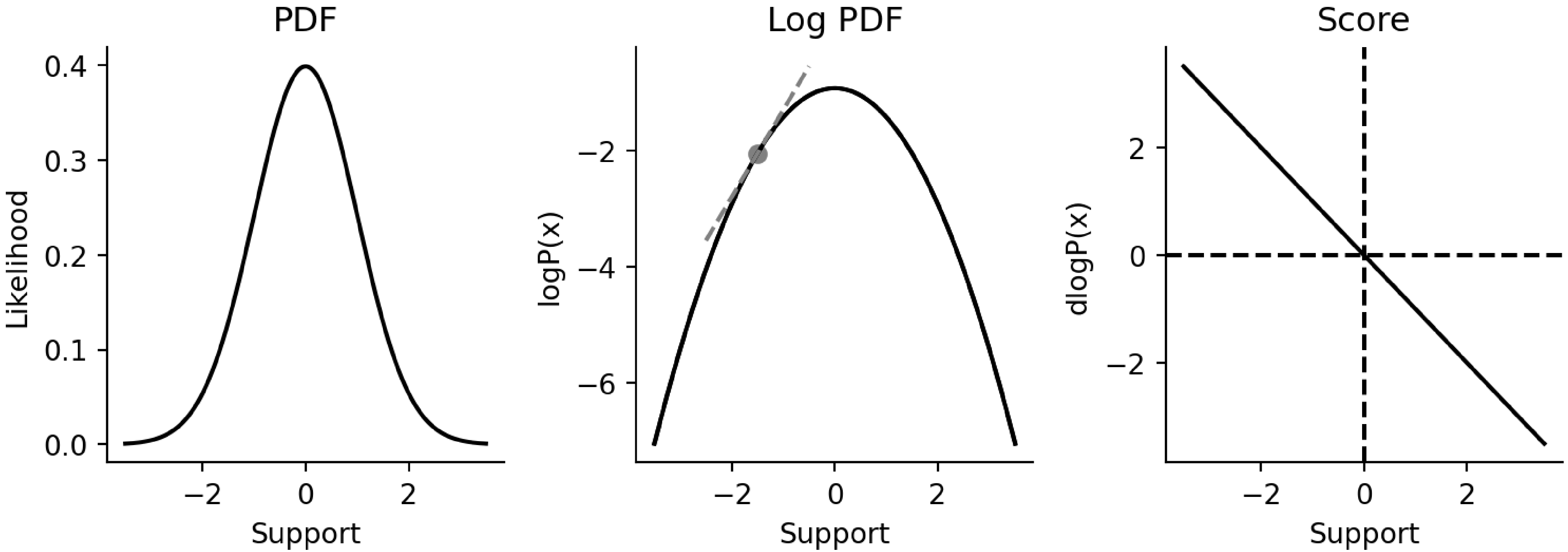
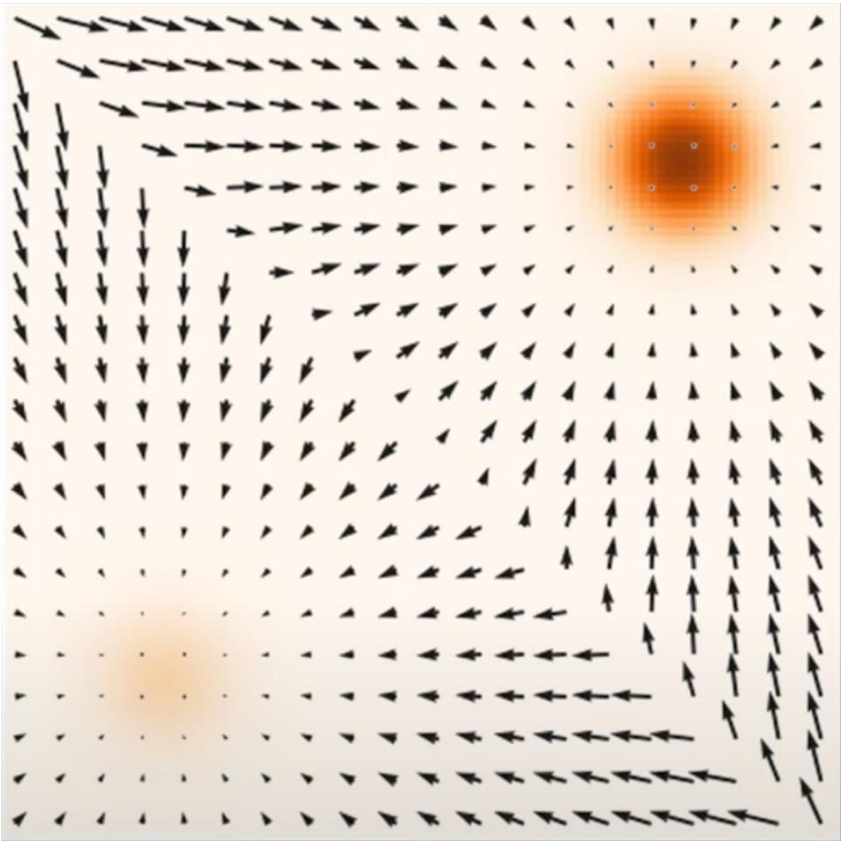


Figure source: “A Pedagogical Introduction to Score Models”, Eric Ma

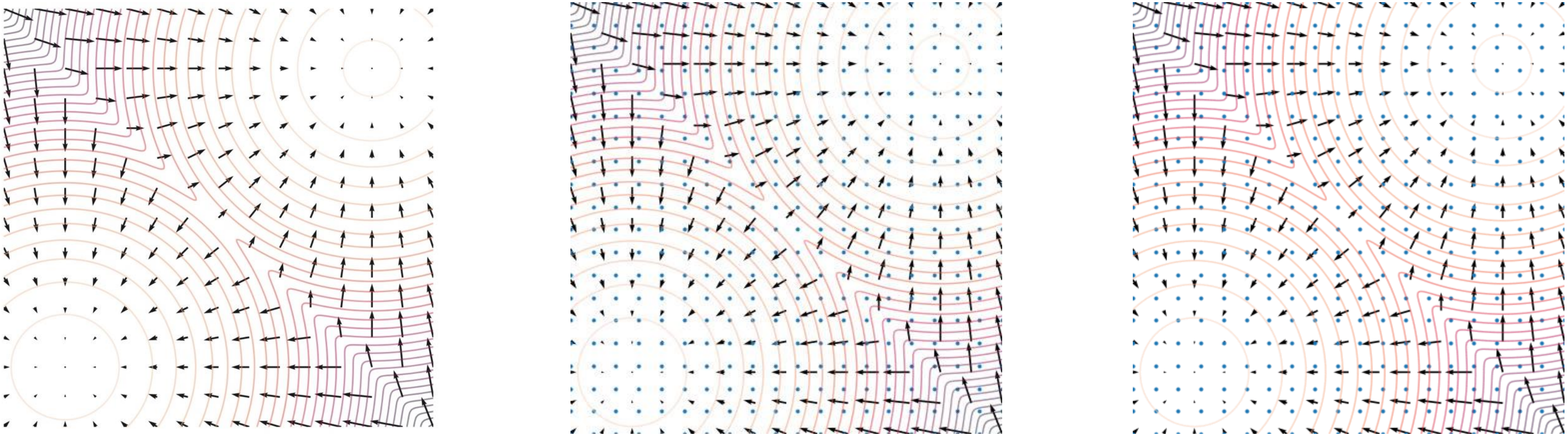
What is “score”?

- Score: gradient of log-likelihood $s(x) = \nabla_x \log p(x)$



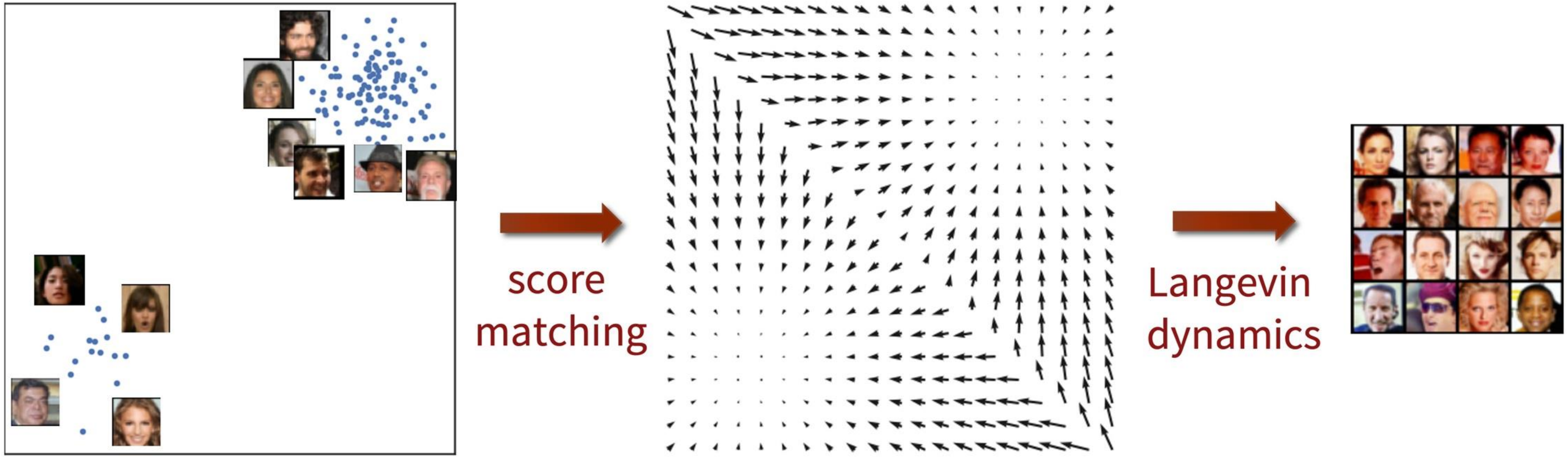
Why is score useful?

- If we know the score, we also know the distribution (implicitly)
- We can also draw samples from the implicit distribution



Langevin dynamics: $x_{i+1} = x_i + \underbrace{\epsilon \nabla_x \log p(x)}_{\text{Move towards the direction that increases likelihood}} + \underbrace{\sqrt{2\epsilon} z_i}_{\text{add random perturbations (Why?)}}$, ϵ is small scalar, $z_i \sim N(0, I)$ has same dimension as x

Why is score useful?



Data samples | Scores | New samples

$\{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \overset{\text{i.i.d.}}{\sim} p(\mathbf{x})$

$\mathbf{s}_\theta(\mathbf{x}) \approx \nabla_\mathbf{x} \log p(\mathbf{x})$

Q: Why estimating the score, instead of the probability density directly?

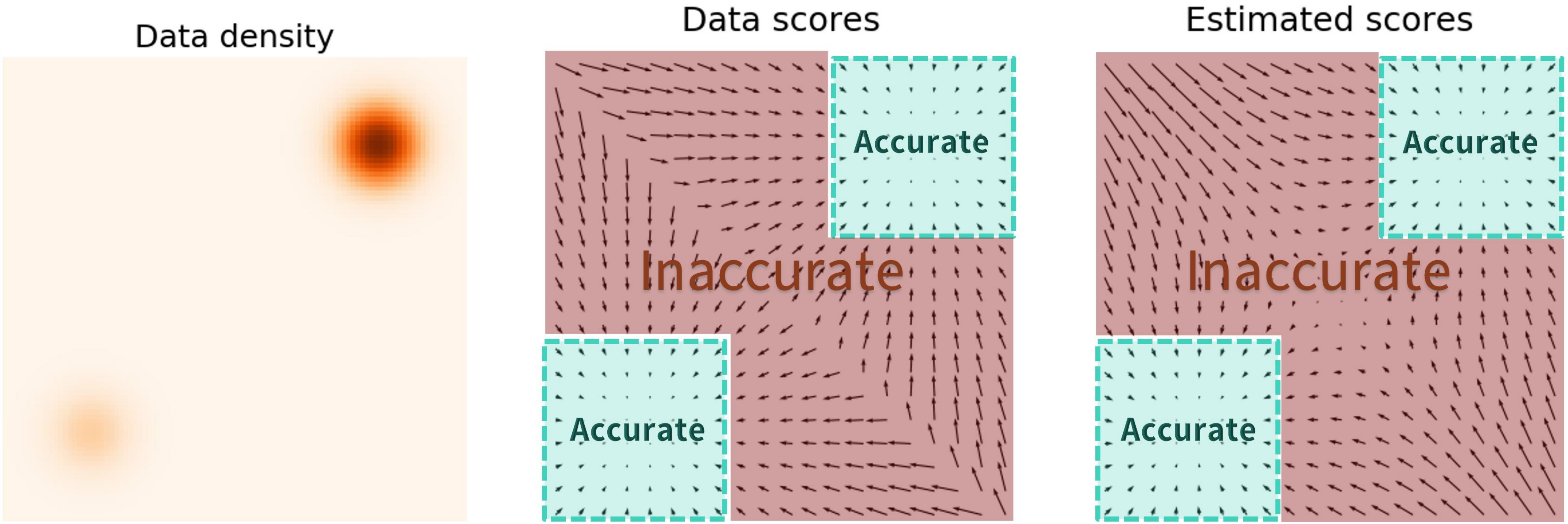
Yang Song, “Generative Modeling by Estimating Gradients of the Data Distribution”

How to estimate the score?

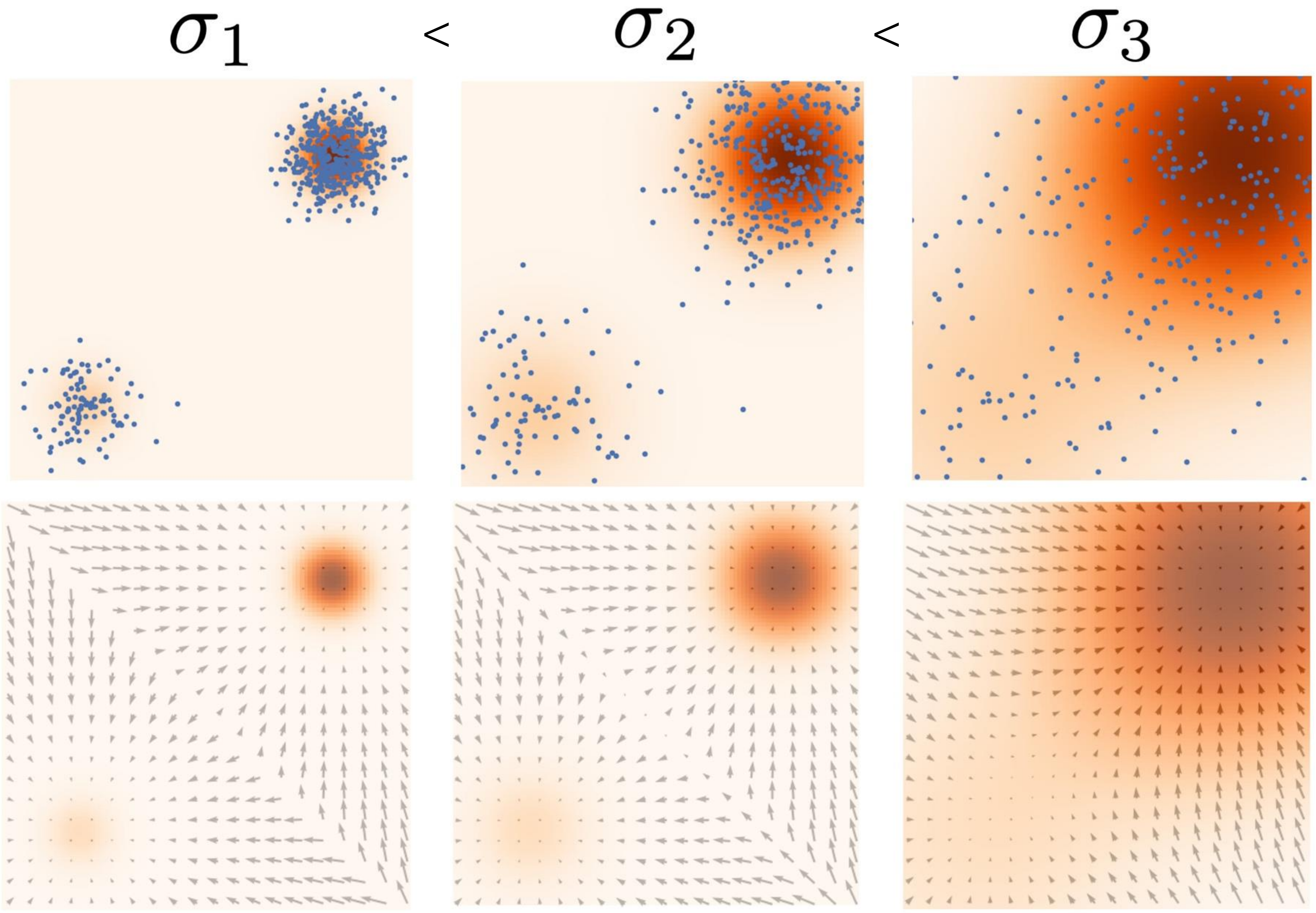
- $\mathbb{E}_{x\sim p(x)} \| s_\theta(x) - \underbrace{\nabla_x \log p(x)}_{\text{☹ We don't know}} \|^2$
- Denoising Score Matching
  - Add Gaussian noise to data: $\tilde{x} = x + \sigma \cdot \epsilon, \epsilon \sim N(0, I)$
  - We can estimate the score of the noised distribution $p(\tilde{x})$

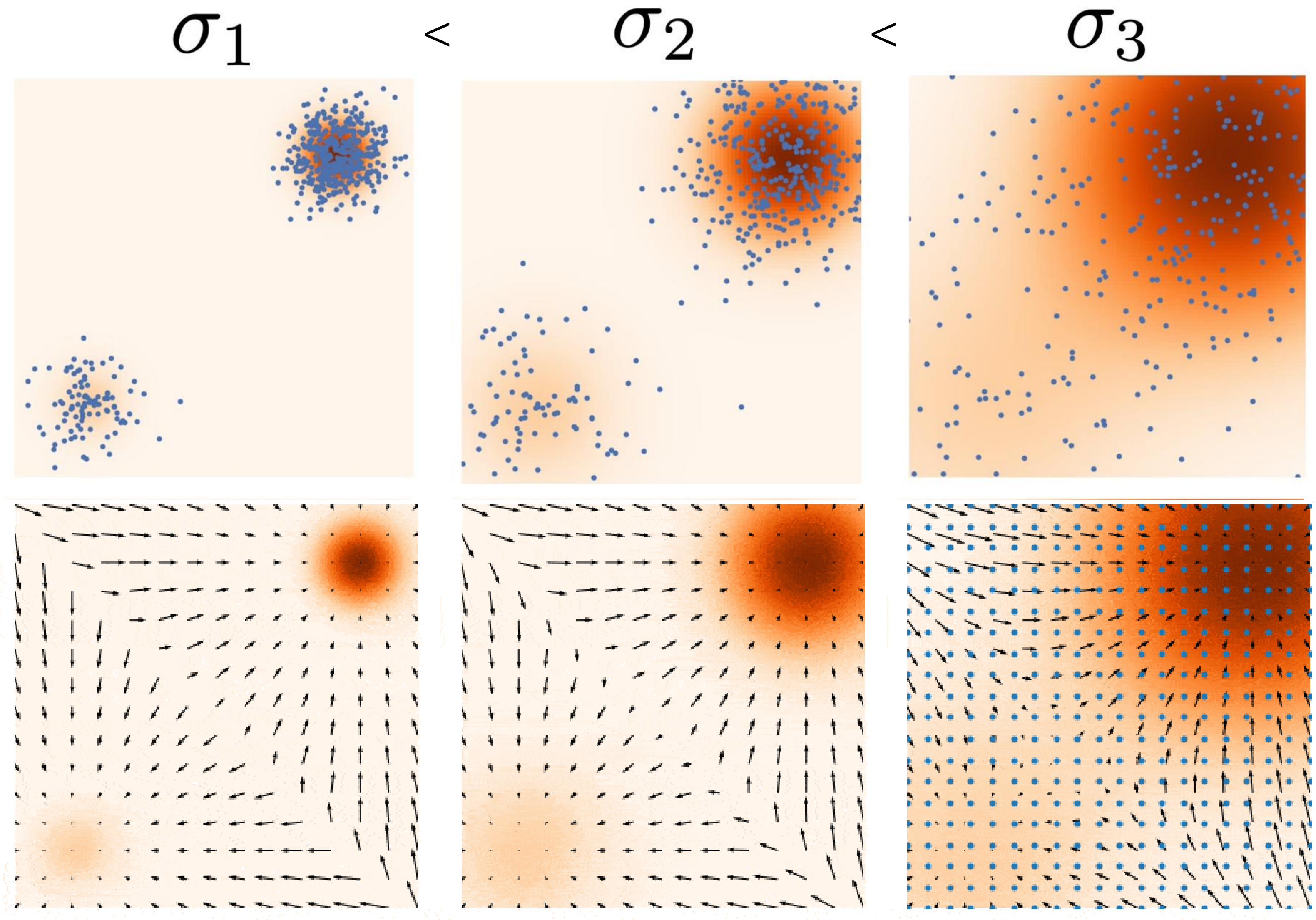
$$\mathbb{E}_{\tilde{x}\sim p(\tilde{x}|x),\, x\sim p(x)} \| s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p(\tilde{x}) \|^2$$
$$= \mathbb{E}_{\tilde{x}\sim p(\tilde{x}|x),\, x\sim p(x)} \| s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p(\tilde{x}|x) \|^2 + \text{Constant}$$

Remember in Diffusion VAE derivation: $\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} \Rightarrow \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1}, x_0)}$



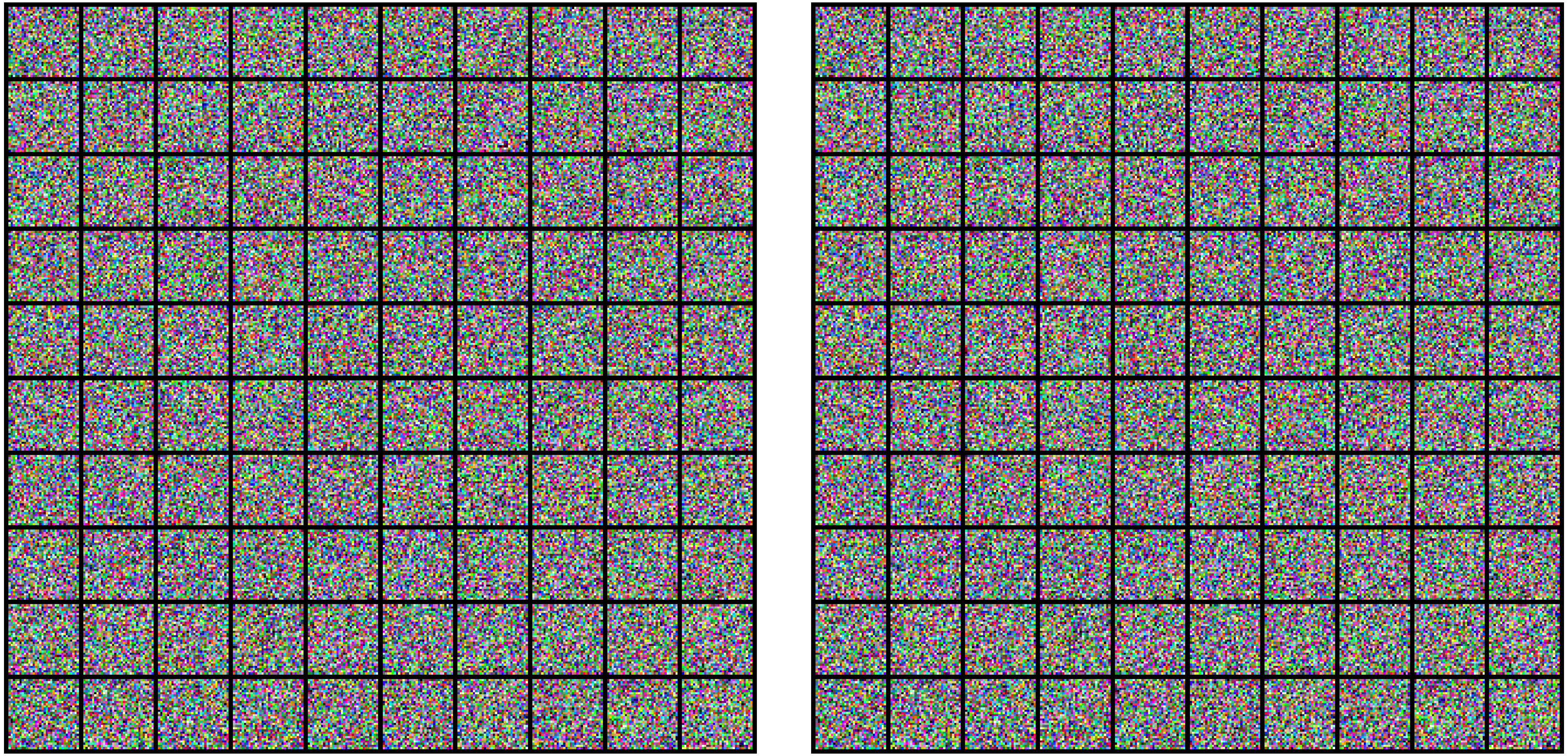
Data density
Data scores
Estimated scores
Accurate
Inaccurate
Accurate
Accurate
Inaccurate
Accurate


$\sigma_1 < \sigma_2 < \sigma_3$



σ_1 < σ_2 < σ_3

Score-based Image Generation

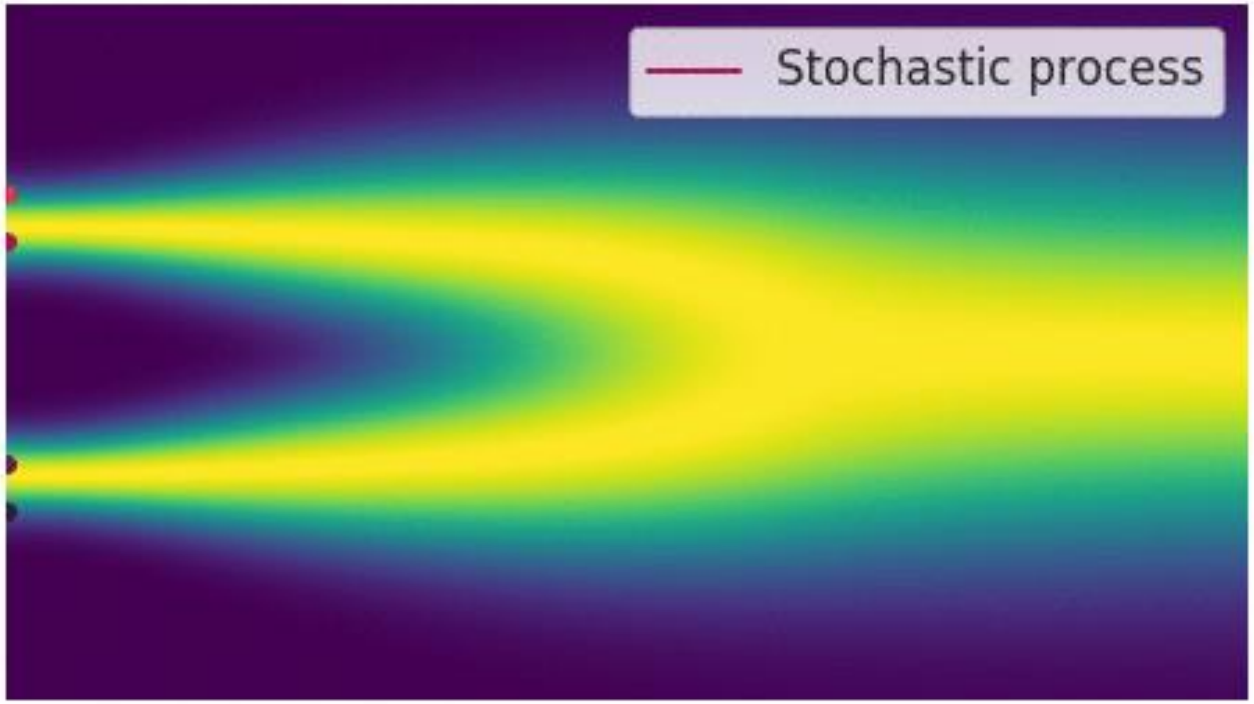


Song and Ermon, “Generative Modeling by Estimating Gradients of the Data Distribution”

Stochastic Differential Equations (SDEs)

$$d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}$$

infinitesimal change in $\boldsymbol{x}$ | infinitesimal change in t | infinitesimal Gaussian noise $d\boldsymbol{w} = N(0, dt\boldsymbol{I})$



Stochastic Differential Equations (SDEs)

Drift coefficient

diffusion coefficient

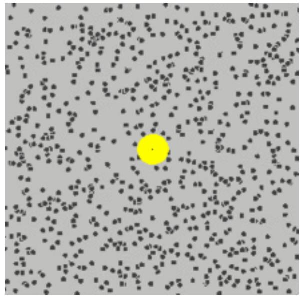
$$d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}$$

infinitesimal change in $\boldsymbol{x}$

infinitesimal change in t

infinitesimal Gaussian noise

$d\boldsymbol{w} = N(0, dt\boldsymbol{I})$



Brownian motion/
Wiener process

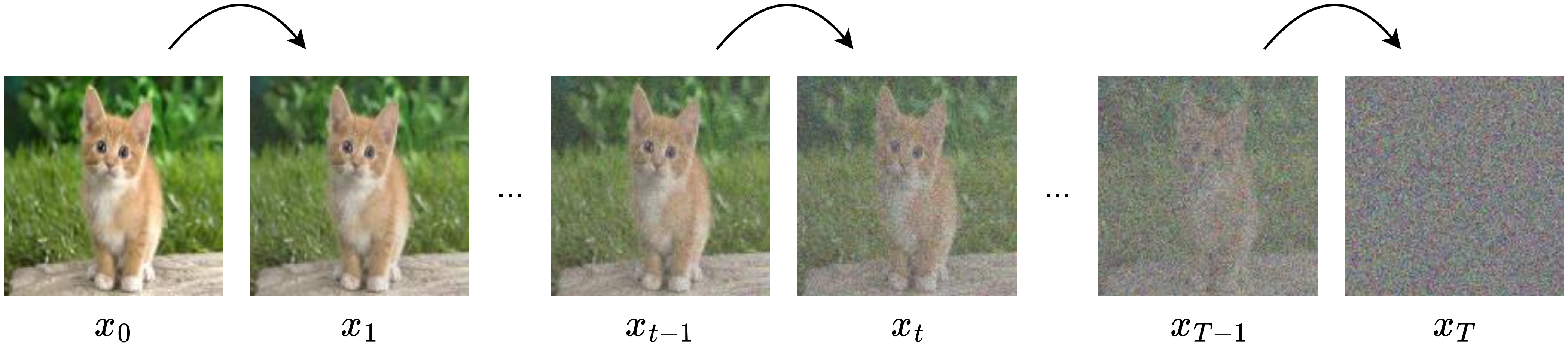
Stochastic Differential Equations (SDEs)

$$d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}$$

Euler method: $\Delta x = f(x, t)\Delta t + g(t)\sqrt{\Delta t}\epsilon$

Diffusion Models with Infinite Steps

- $x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t}\epsilon, \epsilon \sim N(0, I)$
- T is usually very large (e.g., 1000)
- What if $T \to \infty$?



Diffusion Models with Infinite Steps

- $x_t = \sqrt{1-\beta_t} x_{t-1} + \sqrt{\beta_t}\epsilon, \epsilon \sim N(0, I)$
- $x_t = \sqrt{1-\beta(t)\Delta t}\, x_{t-\Delta t} + \sqrt{\beta(t)\Delta t}\, \epsilon, \epsilon \sim N(0, I)$

Stochastic Differential Equations (SDEs)

$$d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}$$

Euler method: $\Delta x = f(x,t)\Delta t + g(t)\sqrt{\Delta t}\epsilon$

Diffusion Model with Infinite Forward Steps: $\Delta x = -\frac{1}{2}\beta(t)\Delta t + \sqrt{\beta(t)\Delta t}\ \epsilon$

Diffusion Model is an SDE!

$f(x,t) = -\frac{1}{2}\beta(t), g(t) = \sqrt{\beta(t)}$

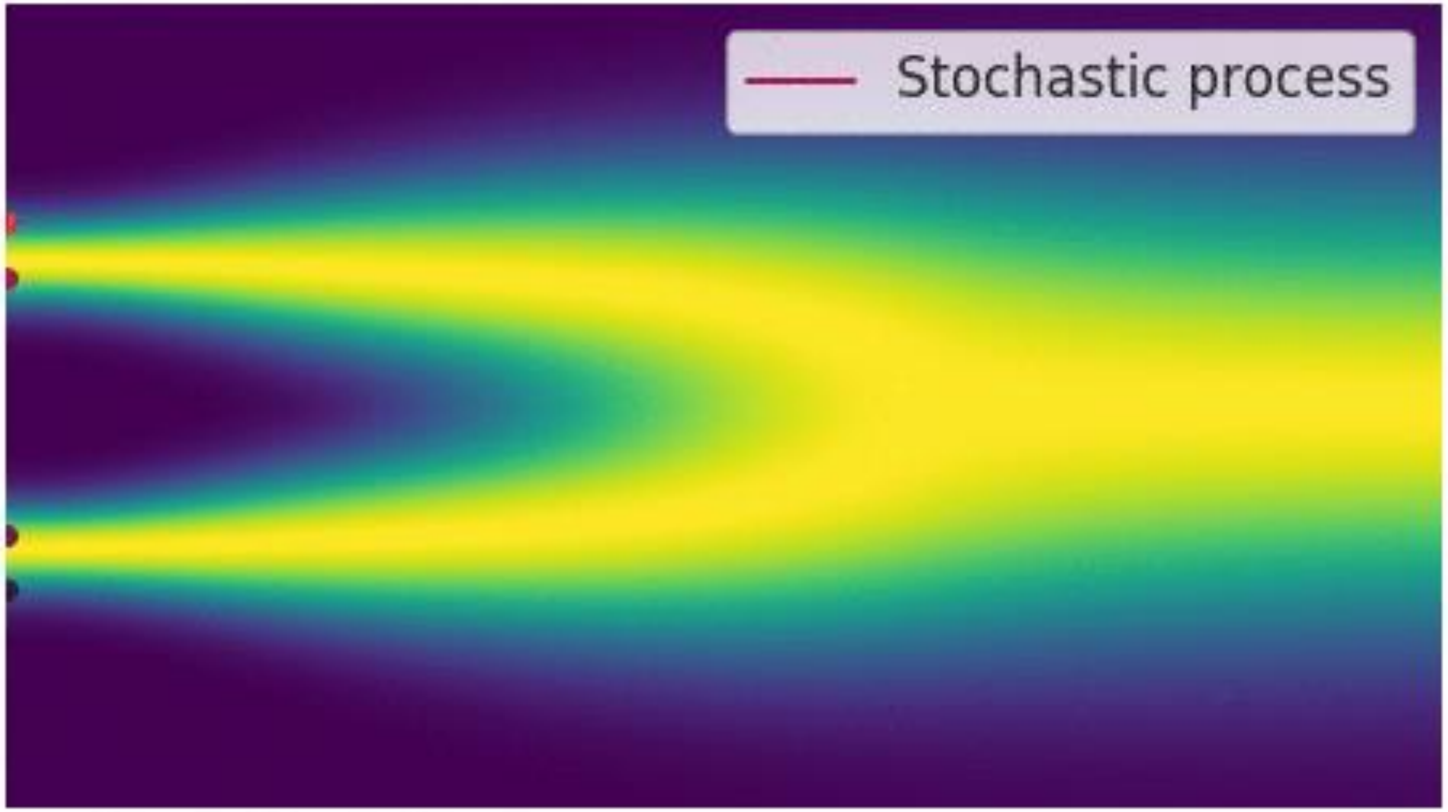


Figure source: Aaron Lou , “Reflected Diffusion Models”

How to reverse an SDE?

- Any SDE has a corresponding **reverse SDE**

How to reverse an SDE?

- Any SDE has a corresponding **reverse SDE**

This seems easy

How to reverse an SDE?

- Any SDE has a corresponding **reverse SDE**

This seems easy

This seems hard

Second law of thermodynamics

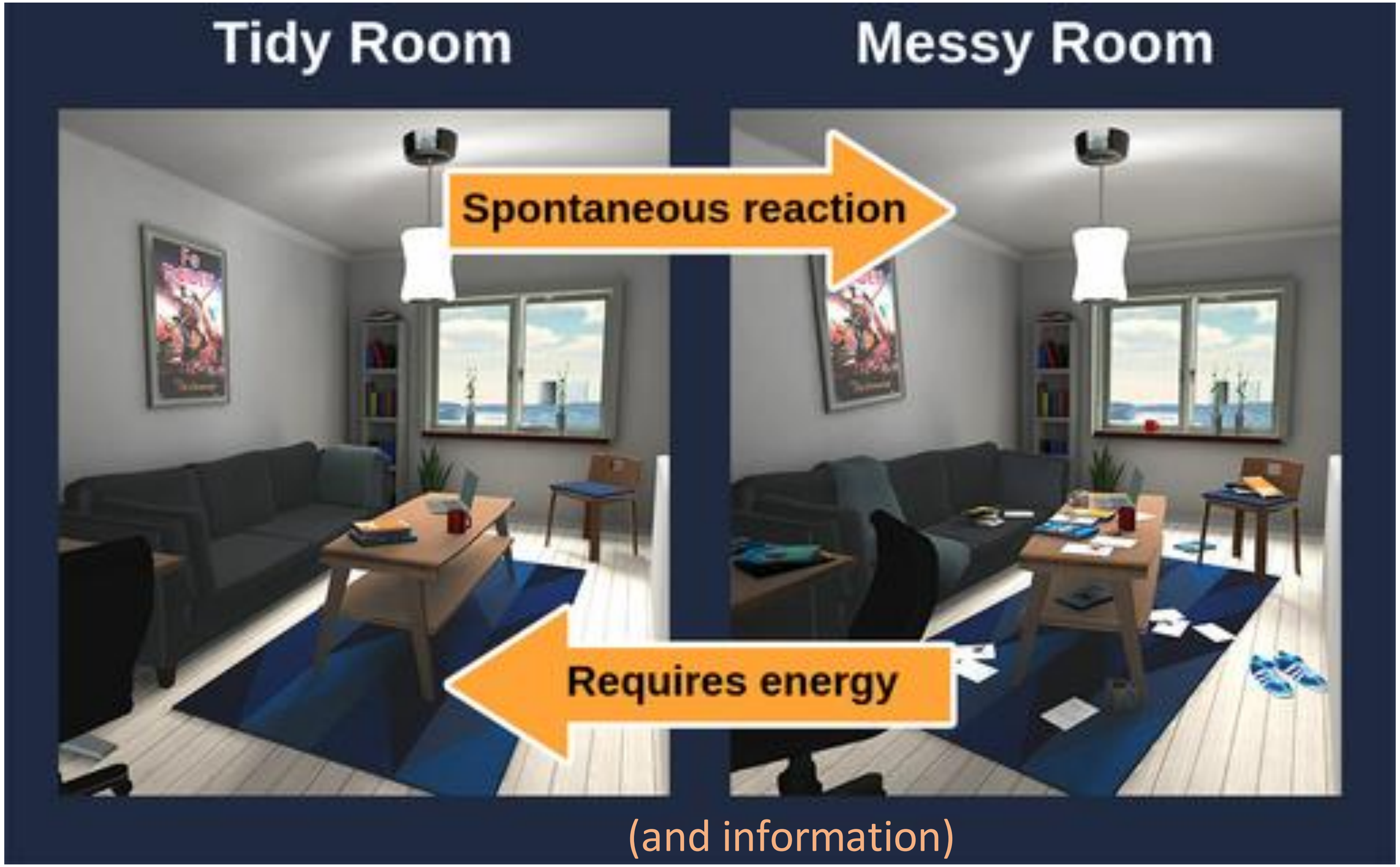
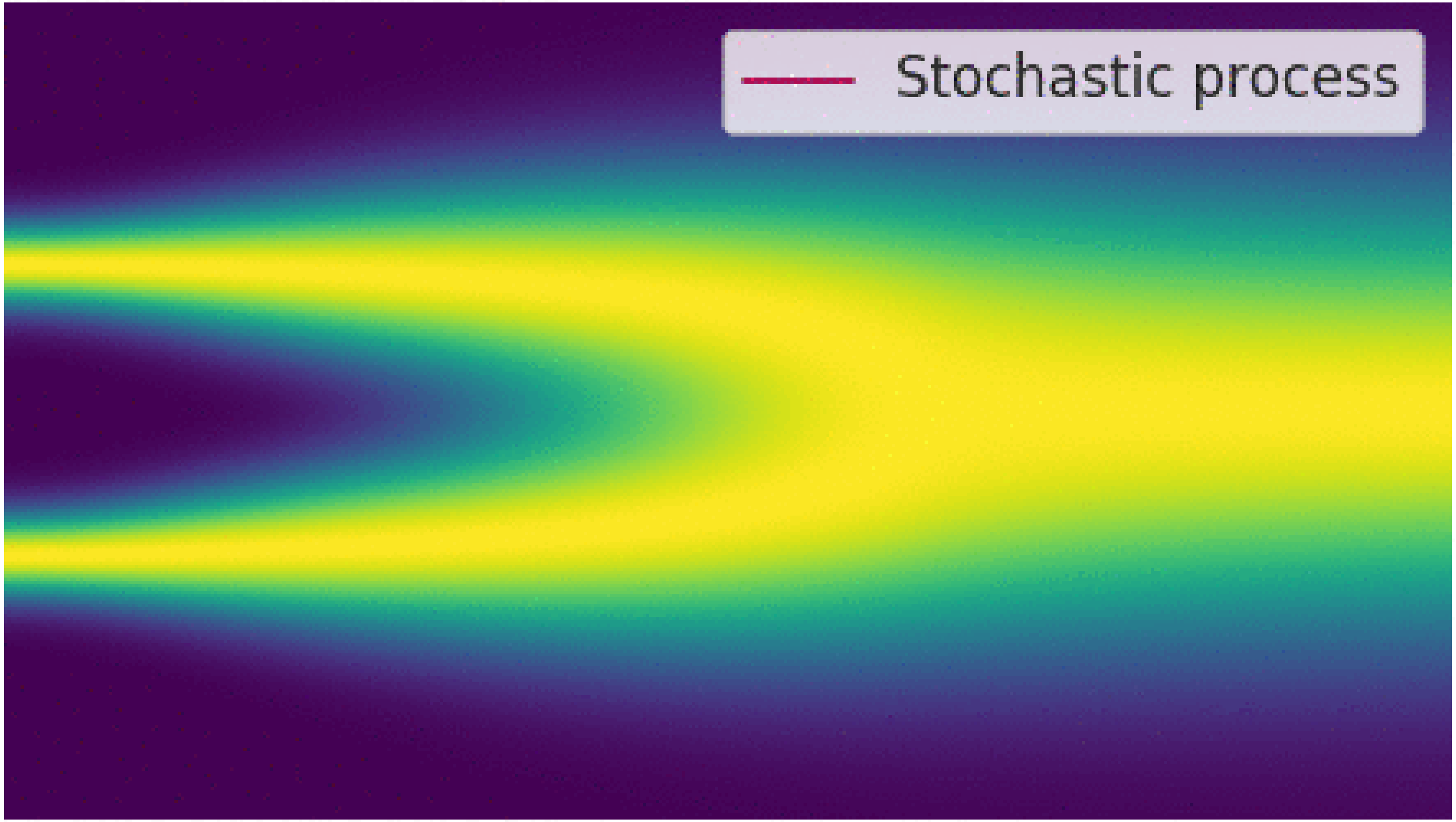


Figure source: “Labster Theory”

How to reverse an SDE?



How to reverse an SDE?

- Any SDE has a corresponding **reverse SDE**

Forward: $d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}$

Reverse: $d\boldsymbol{x} = (f(\boldsymbol{x}, t) - g(t)^2 \nabla_x \log p_t(x))dt + g(t)d\bar{\boldsymbol{w}}$

Score of the noisy distribution

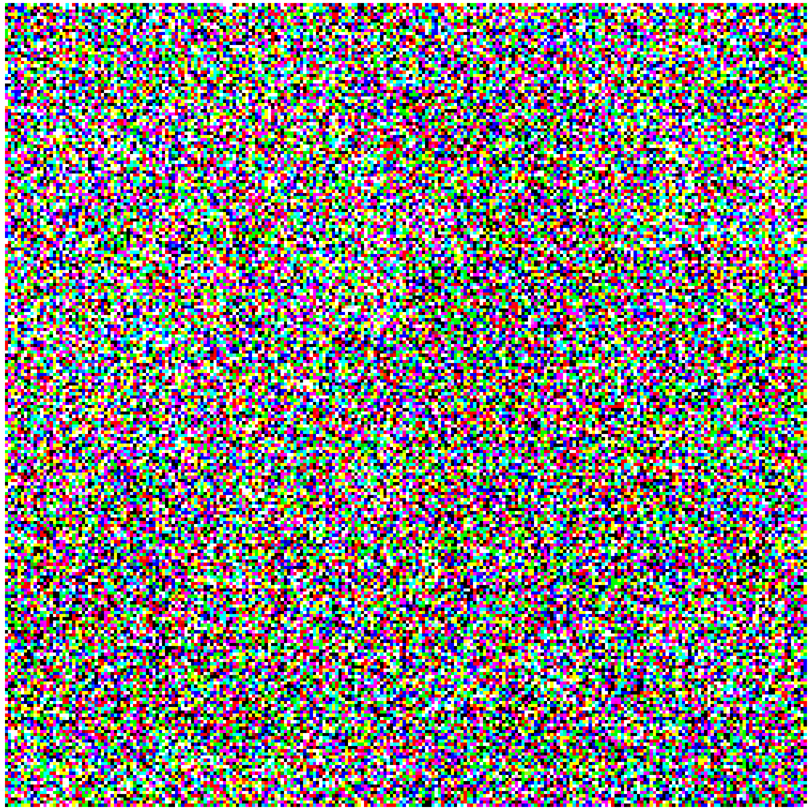
Learning the reverse SDE = Learning the score = Learning to denoise


NYPD
SCORE MATCHING
DENOISING DIFFUSION
imgflip.com

Sampling an image = solve the reverse SDE

Reverse SDE: $d\boldsymbol{x} = (f(\boldsymbol{x}, t) - g(t)^2 \underbrace{\nabla_x \log p_t(x)}_{\approx s_\theta(x)})dt + g(t)d\bar{\boldsymbol{w}}$

- Numerical, discretized SDE solvers
- Sample $x \sim p_T(x)$, set $t = T$, $\Delta t = -T/N$ (N is the number of steps)
- While $t > 0$:
  - $\Delta x = \big(f(\boldsymbol{x}, t) - g(t)^2 s_\theta(x)\big)\Delta t + g(t)\sqrt{\Delta t}\epsilon, \epsilon \sim N(0, I)$
  - $x = x + \Delta x$
  - $t = t + \Delta \mathrm{t}$



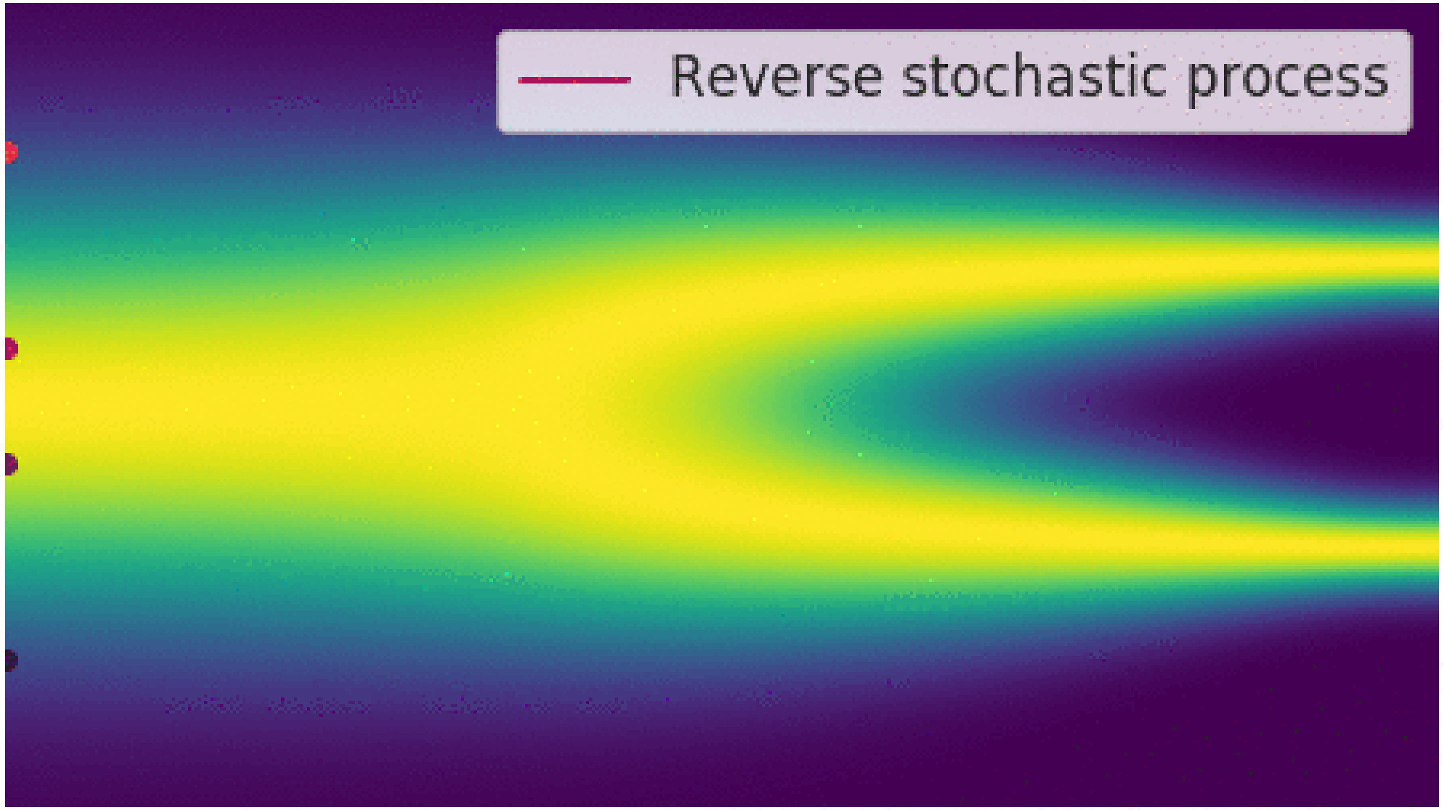
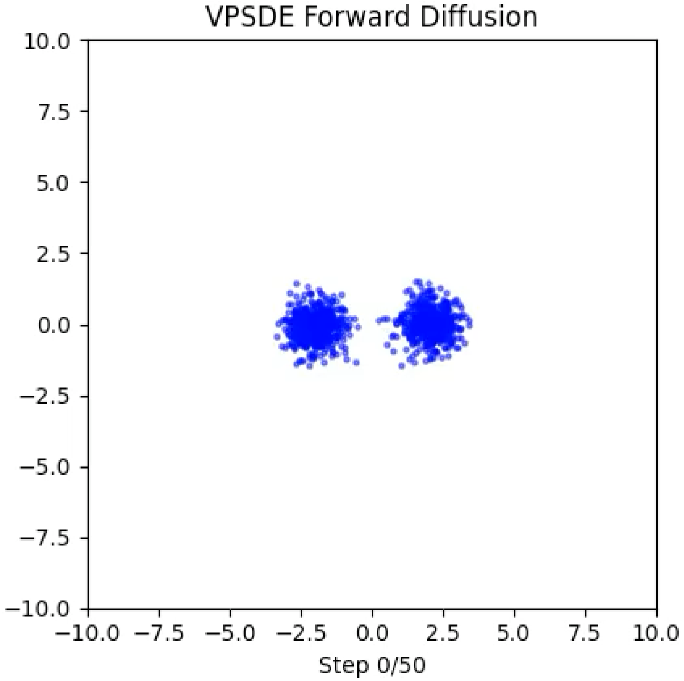


Figure source: Aaron Lou , “Reflected Diffusion Models”

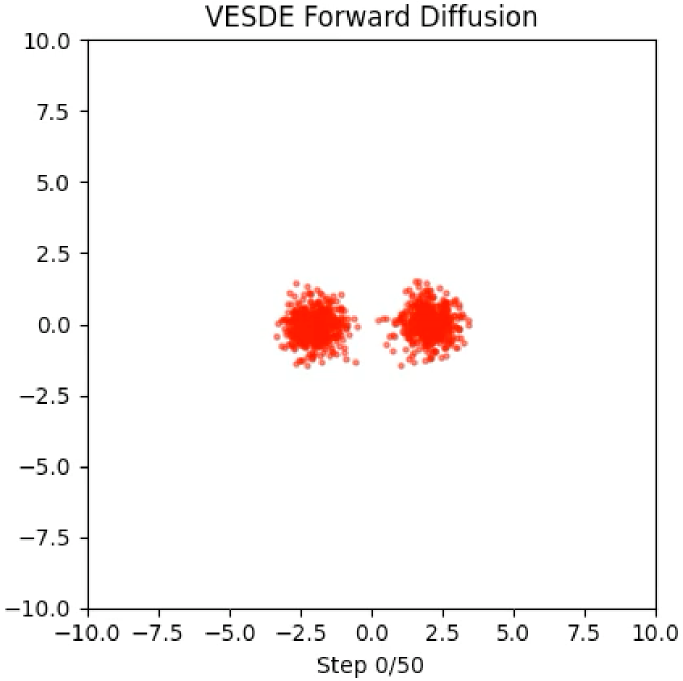
Compare SDE vs. VAE interpretation. What advantages did we get?

- Generalize to arbitrary SDEs $d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}$
  - Variance preserving (VP) and variance exploding (VE) SDE



VPSDE: Add noise while attenuating data

Example: $f(x, t) = -\frac{1}{2}\beta(t), g(t) = \sqrt{\beta(t)}$



VPSDE: Add noise without attenuating data

$f(x, t) = 0, g(t) = \sqrt{\beta(t)}$

Benefits of SDE interpretation (vs. VAE)

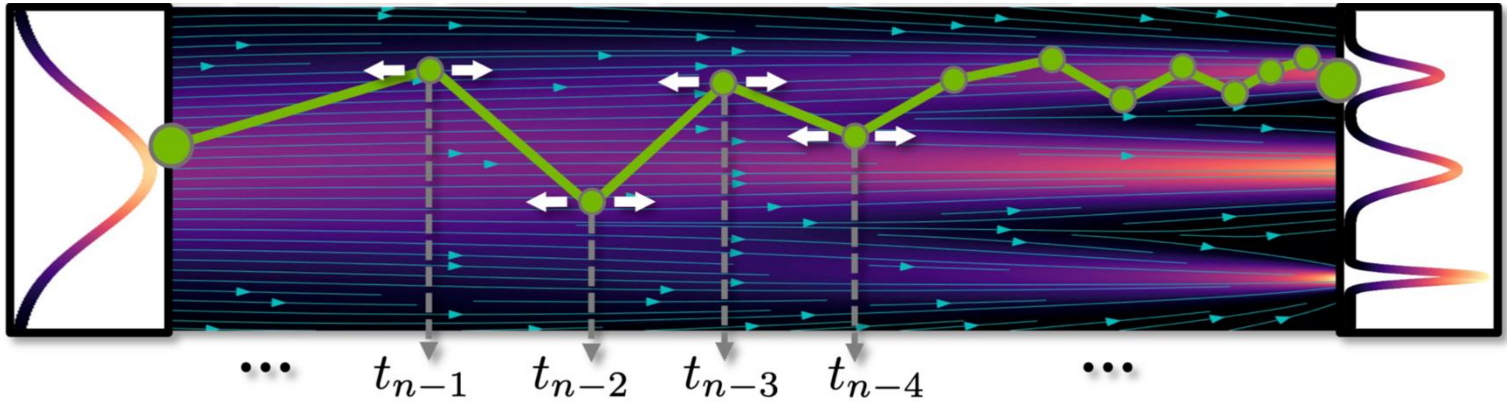
- Generalize to arbitrary SDEs $d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}$
  - Variance preserving (VP) and variance exploding (VE) SDE
- Decoupled training and inference
  - Training: estimate the **score** at various **noise levels** $\nabla_x \log p_\sigma(x)$
    - Don’t care about SDE, discretization, etc..
    - Don’t even care about “time”
    - Sample a noise level (σ) from some continuous distribution, add noise, denoise

Benefits of SDE interpretation (vs. VAE)

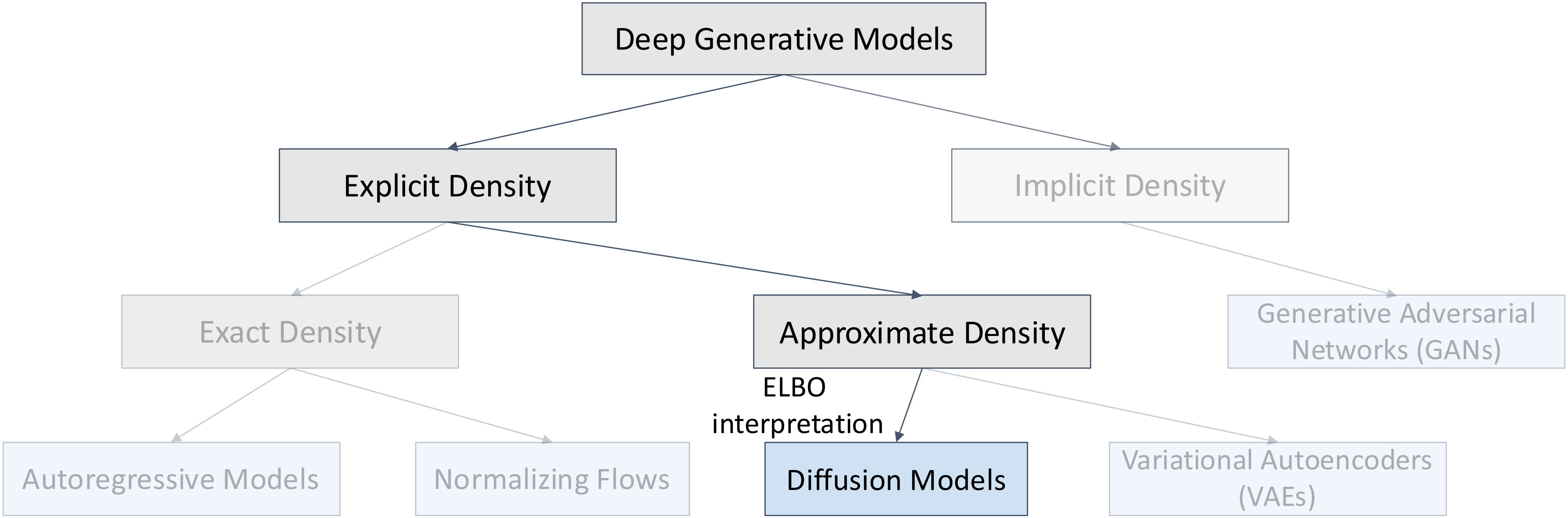
- Generalize to arbitrary SDEs $d\boldsymbol{x} = f(\boldsymbol{x}, t)dt + g(t)d\boldsymbol{w}$
  - Variance preserving (VP) and variance exploding (VE) SDE
- Decoupled training and inference
  - Training: estimate the **score** at various **noise levels** $\nabla_x \log p_\sigma(x)$
  - Inference: solve an SDE
    - Flexible number of function evaluation (NFE)
    - More advanced solvers

1000 steps

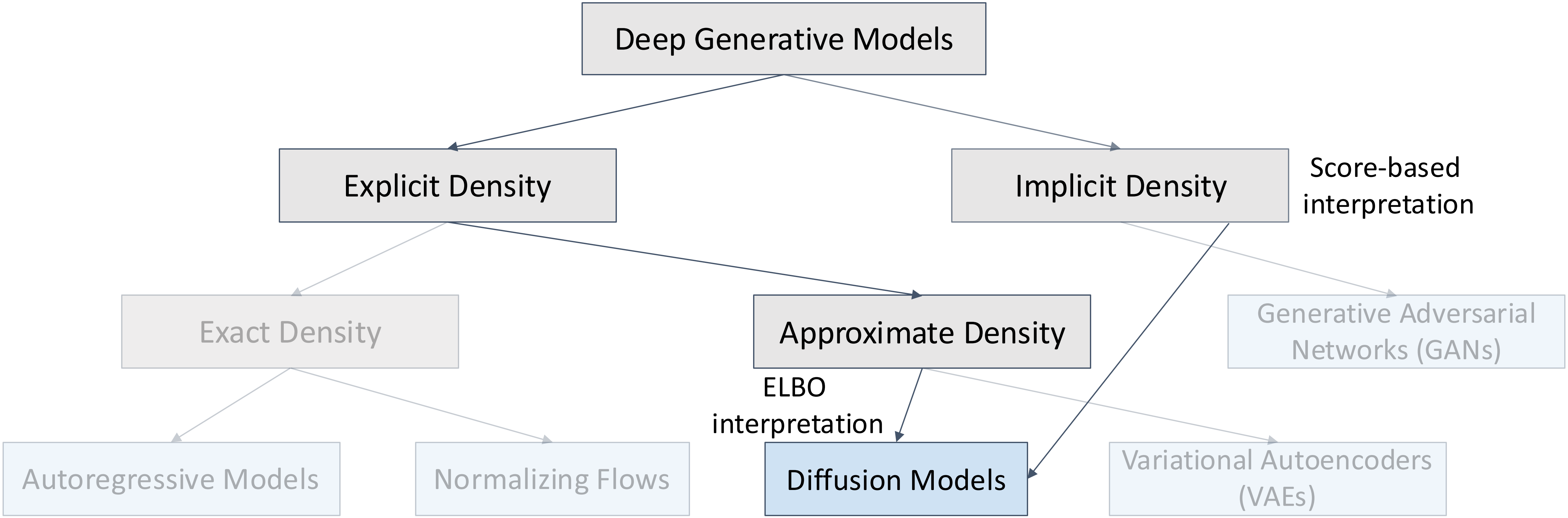
200 steps



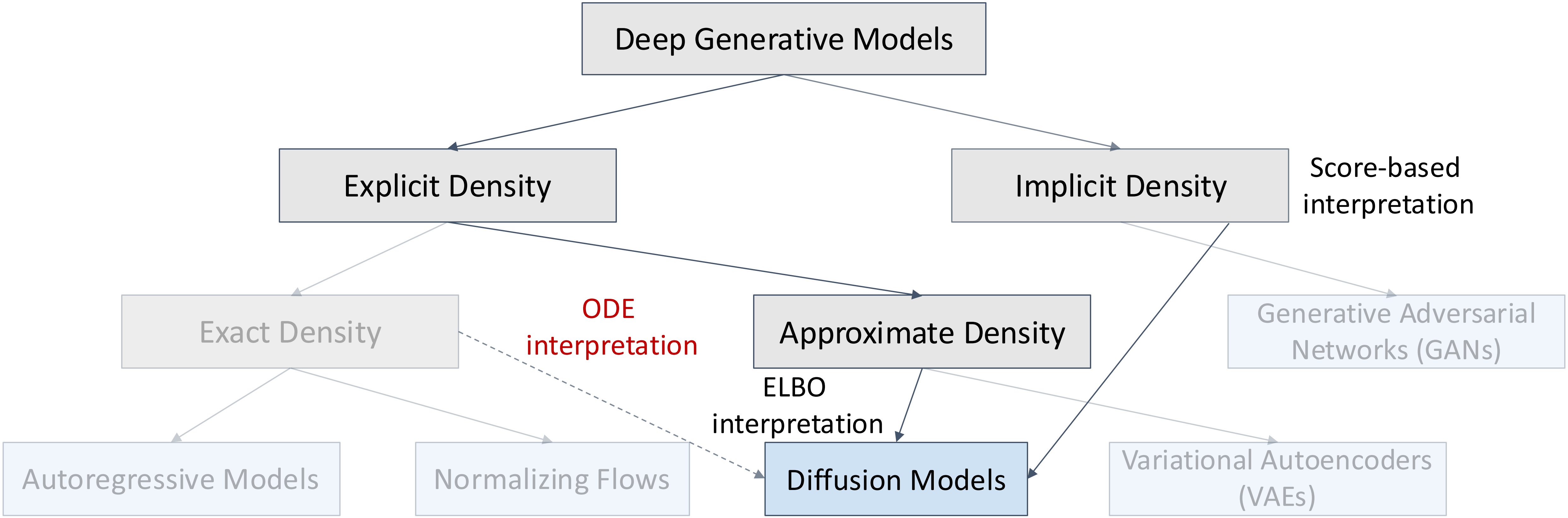
Diffusion Models (continue...)



Diffusion Models (continue...)



Diffusion Models (continue...)



5 Minute Quiz

- On Canvas
- Passcode: elephant